ENGINEERING Information Technology & Electronic Engineering
https://www.academax.com/EITEE; https://www.engineering.org.cn
https://ieeexplore.ieee.org/xpl/RecentIssue.jsp?punumber=11344065
E-mail: eng.itee@zju.edu.cn

Engineering Journals
Available on IEEE Xplore®

Research Article
https://doi.org/10.1631/ENG.ITEE.2026.0044

Three-dimensional affordance segmentation for object point cloud driven by language instructions

Jiaxuan DU[1], Hao WU[1]✉, Qing MA[1], Guohui TIAN[1], Zhixian ZHAO[1], Shuwen LENG[2]

[1]*School of Control Science and Engineering, Shandong University, Jinan 250061, China*
[2]*Shandong Branch of China Huaneng Group Co., Ltd., Jinan 250014, China*

Abstract: The location where a robot grasps an object is closely related to the task type. For the same object, different user requirements may necessitate different grasping strategies. Visual affordance serves as a reliable source of prior knowledge for manipulation. Existing methods learn affordance from images or videos, but planar affordance lacks the spatial information required for 6-degree-of-freedom (6-DoF) manipulation. Furthermore, current approaches are limited to affordances associated with predefined categories and cannot directly infer affordances from user instructions. To address such limitations, we propose a novel task: instruction-driven three-dimensional (3D) object affordance segmentation. To support this research, we introduce an instruction–affordance dataset (IAD), a challenging dataset consisting of 7190 object instances across 20 common object categories, paired with 624 manipulation instructions that specify the corresponding affordances. To evaluate generalization to novel commands, our dataset includes both seen and unseen settings. Building on this, we design an instruction-driven 3D affordance segmentation (IDAS) network, which extracts point cloud features and integrates instruction features layer by layer. Given a user instruction, our method segments suggested manipulation regions on the object's point cloud, thereby guiding the selection of optimal grasp poses. Experimental results show that our method outperforms other related approaches under both seen and unseen settings, demonstrating generalization ability to diverse user commands and unknown affordances.

Key words: Visual affordance; Point cloud segmentation; Open vocabulary; Multimodal fusion; Service robot

1 Introduction

Service robots require intelligence to accomplish tasks assigned by users (Wang et al., 2022). After actively approaching the target object (Liu SP et al., 2022), the robot needs to analyze how to manipulate it. Manipulation strategies for service robots must align with human expectations regarding how they interact with objects. For instance, when picking up a cup, humans generally prefer that the robot grasp the body of the cup instead of the rim, which is the part that comes into contact with their mouth. This preference reflects a more acceptable and hygienic approach to service. Additionally, the appropriate grasping location on an object often varies depending on user requirements. For example, when the task is to open a backpack, the robot needs to operate the zipper. However, if the task is to lift a backpack, it should instead grasp the handle.

Affordance acts as an informative prior that facilitates decision-making in robot service. Affordance refers to what the environment offers to an animal, including the opportunities it provides for interaction (Gibson, 1978). In the context of robot manipulation, affordances describe how objects present action possibilities for human interaction. These affordances are crucial for understanding how to manipulate objects effectively in robotics, and have been the subject of extensive study. Early research primarily focused on learning affordances from static images (Song et al., 2015; Roy and Todorovic, 2016; Do et al., 2018; Ardón et al., 2019) or from videos of human–object interaction (Fang K et al., 2018; Nagarajan et al., 2019; Goyal et al., 2022). More recent studies (Deng et al., 2021; Nguyen et al., 2023; Yang YH et al., 2023; Li YC et al., 2024) have advanced the concept by extending affordance segmentation into the three-dimensional (3D) domain using point cloud data. However, these affordance learning methods do

✉ Hao WU, wh911@sdu.edu.cn
Jiaxuan DU, https://orcid.org/0009-0001-0930-9958
Hao WU, https://orcid.org/0000-0001-6993-8863
Qing MA, https://orcid.org/0000-0002-3902-3635
Guohui TIAN, https://orcid.org/0000-0001-8332-3064
CLC number: TP242.62

Received: Feb. 6, 2026; Revision accepted: Mar. 22, 2026;
Crosschecked: Apr. 7, 2026

not directly provide knowledge for manipulation; rather, they offer a broader conceptual understanding of the environment, which is difficult to apply in practice.

With the prompt of affordance, grasping approaches can provide more task-oriented and spatially reasonable grasp poses. Traditional grasp-planning approaches (Mousavian et al., 2019; Sundermeyer et al., 2021; Fang HS et al., 2023) primarily focus on physical feasibility, aiming to maximize the success rates of grasps without taking into account the humanlikeness or task relevance of the grip. As a result, these general-purpose methods are usually limited to simple manipulation tasks (Qin et al., 2023), such as sequentially picking up and placing objects on a tabletop. Therefore, learning grasp-relevant affordances is essential for handling diverse tasks.

Understanding natural language instructions is essential for intelligent robots, as it allows nonexpert users to interact with them intuitively. For robots to accomplish this, they need to interpret user instructions and devise appropriate manipulation strategies. With the emergence of large-scale language models (Devlin et al., 2019; Radford et al., 2021), vision–language multimodal learning (Yang ZY et al., 2021; Roh et al., 2022) has been significantly advanced.

Despite recent advancements, open-vocabulary affordance segmentation of 3D object point clouds continues to be a significant challenge. Directly inferring affordances from user instructions remains an open research problem. To help robots learn the relationship between user instructions and object affordances and thereby improve their manipulation performance, we propose a novel approach called instruction-driven 3D affordance segmentation (IDAS). Using the general knowledge encoded in pre-trained language models, our method demonstrates generalization capabilities even with limited training data. Our key contributions are summarized as follows:

1. We present a novel task and instruction-driven affordance segmentation on object point clouds, as depicted in Fig. 1. To support this research, we have created a new dataset called the instruction–affordance dataset (IAD). This dataset contains 14 387 instruction–object pairs, and includes 20 common household objects and 12 affordance types that are relevant to manipulation tasks.

2. We introduce an IDAS network. Our proposed query-aware modulation (QAM) module and verb–noun attention (VNA) module demonstrate a strong capability to align with instruction semantics with point cloud features.

3. Our method demonstrates superior performance compared to existing approaches in both seen and unseen instruction–affordance settings, highlighting its ability to generalize across diverse user commands.

2 Related works

2.1 Visual affordance in robot manipulation

Affordance segmentation extends beyond traditional visual tasks, such as object detection and semantic segmentation, by providing a more profound understanding of object functionality. A significant amount of research has concentrated on supervised learning approaches that map objects to their respective affordance regions (Song et al., 2015; Roy and Todorovic, 2016; Do et al., 2018; Fang K et al., 2018; Ardón et al., 2019; Nagarajan et al., 2019; Goyal et al., 2022). For example, Do et al. (2018) proposed pixel-level affordance segmentation on images. However, two-dimensional (2D) methods are inherently limited by their dependence on planar information, which often neglects the geometric structure of objects. This limitation makes 2D methods less effective for applications that require direct guidance for manipulation. Qian et al. (2024) used vision–language models for 2D visual affordance grounding, but their results highlight the limitations of relying exclusively on 2D data, and underscore the need to incorporate 3D information into affordance grounding. Deng et al. (2021) introduced the first affordance segmentation dataset for object point clouds. Subsequent works, such as Nguyen et al. (2023) and Li YC et al. (2024), explore open-vocabulary 3D affordance segmentation, while Mo et al. (2022) investigated affordance transfer across different 3D object models. In contrast, our research focuses specifically on affordances associated with human–object interaction.

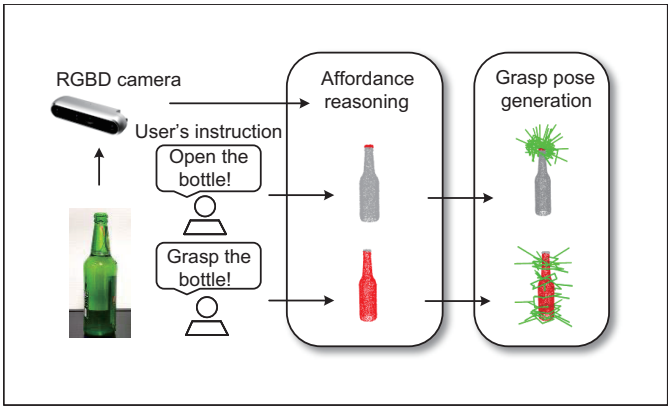

Fig. 1 Overview of the IDAS. Our method can derive distinct affordance region segmentations under different task instructions, which are then used for selecting grasping poses

2.2 Language-guided point cloud segmentation

Language and point cloud multimodal learning has emerged as a crucial task for enabling interactive and intelligent agents to perceive 3D environments based on human natural language. Early works (Achlioptas et al., 2020; Chen et al., 2020) introduce 3D visual grounding benchmarks by aligning free-form referring expressions with target objects in indoor scenes. These methods typically adopt a two-stream architecture, encoding 3D geometric features and language embeddings separately, followed by late fusion via attention mechanisms or matching modules. To improve cross-modal alignment, later methods incorporate Transformer-based fusion (Zhao et al., 2021; Roh et al., 2022), contrastive objectives (Huang et al., 2021), or graph-based relational reasoning (Yang ZY et al., 2021), which allow more fine-grained semantic matching between textual cues and 3D structures. While effective in the context of object-level grounding, these approaches primarily focus on identifying discrete objects rather than segmenting functionally relevant regions. Compared to existing works, our approach tackles a novel setting of instruction-driven

affordance segmentation, where the robot must segment those specific manipulable regions in the point cloud, which fulfill a language-described intent.

3 Dataset

3.1 Collection

The 3D AffordanceNet dataset, proposed by Deng et al. (2021), includes 23 object categories and annotations for 18 types of affordances. Each point cloud representing an object may exhibit multiple affordances. Building on the 3D AffordanceNet, we create a subset by removing objects that are not commonly involved in daily manipulation tasks (e.g., beds) and excluding affordances that do not imply manipulation (e.g., sit).

3.2 Instruction generation

Inspired by Li YC et al. (2024), we generate 16 manipulation-related instructions for each object–affordance pair to serve as a training resource. These instructions are synthesized using 16 custom-designed templates that encompass a diverse range of linguistic styles. This includes polite requests, imperative commands, interrogative forms, and more. We use synonymous expressions to describe the same affordance, and additional elements—such as action goals, methods, or contextual details—are included to enhance semantic variety. Examples of these templates are illustrated in Table 1. To improve efficiency and reduce the manpower required, we prompt generative pre-trained Transformer 4o (GPT-4o) (Islam and Moushi, 2025) with the following request: "Please generate user instructions for each object–affordance pair. The instruction must include the specified object name and be related to the corresponding affordance. Use multiple sentence patterns to create diverse commands." The ratio between human-made instruction and large language model (LLM)-generated instruction is 1:1.

Table 1 Instruction templates

Number	Template
1	Do ...!
2	Can you do ...?
3	Could you do ...?
4	Would you do ...?
5	I want you to do ...
6	Will you please do ...?
7	I'd like you to do ...
8	I hope you do ...
9	Please do ...
10	I need to do ...
11	Do ..., please.
12	Replace action-related terms with their synonyms.
13	Add the purpose of the command.
14	Provide hints about how to perform the action.
15	Add contextual scene details.
16	Provide details of the task requirement.

3.3 Statistical analysis

The process of building the dataset is illustrated in Fig. 2. In our dataset, half of the instructions are manually created, while the other half is generated by LLMs that have been prompted on our designated samples. All generated instructions are manually reviewed to ensure that they align with predefined standards. The number of entries for each object category and each affordance category is presented in Fig. 3a. The instruction length ranges from 4 to 14 words, with detailed statistics provided in Fig. 3b. In addition, the overall vocabulary size is 379, reflecting a diverse set of linguistic expressions across the dataset.

To assess the method's ability to generalize to diverse instructions and objects, we split the dataset into a seen condition and an unseen condition. Under the seen condition, the objects and their corresponding affordances are shared between the training and the testing sets. Conversely, in the unseen condition, the object–affordance pairs that are used for testing are entirely absent in the training data. This setup allows us to evaluate the model's robustness in zero-shot generalization scenarios.

4 Methods

4.1 Problem statement

Our goal is to enable robots to directly analyze the operational affordance regions from user instructions. Considering the lack of existing datasets for instruction-to-affordance mapping, we introduce a new dataset built upon 3D AffordanceNet

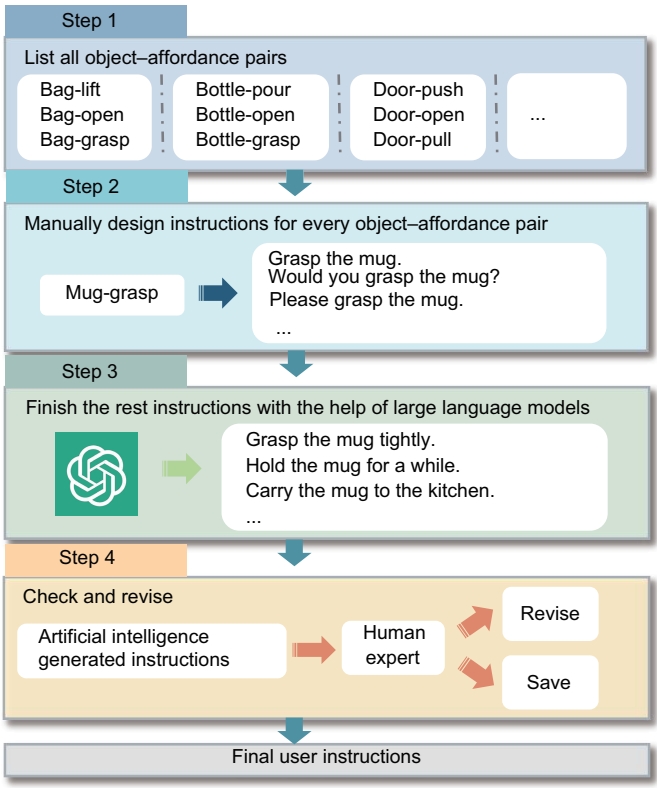

Fig. 2 IAD building process

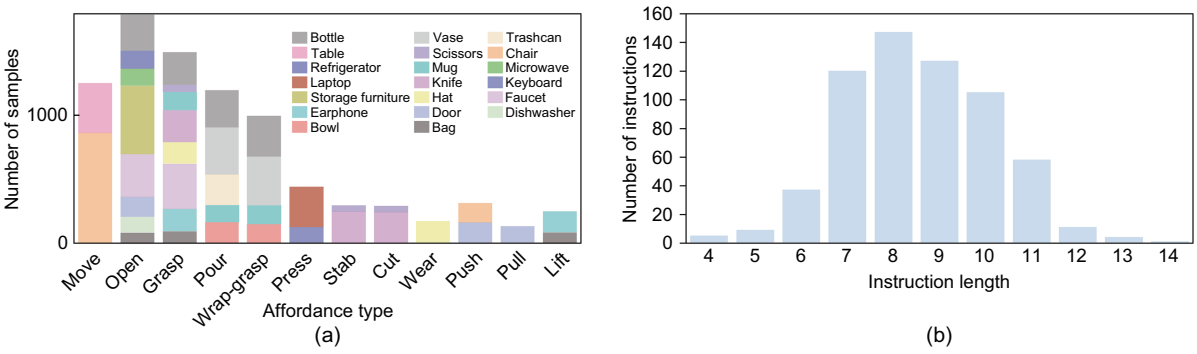

Fig. 3 IAD statistics: (a) affordance type; (b) instruction length

in the previous section. Consider an object point cloud $P = \{\boldsymbol{p}_1, \boldsymbol{p}_2, \ldots, \boldsymbol{p}_n\}$ of n unordered points $\boldsymbol{p}_i$, where $i = 1, 2, \ldots, n$. Each point is represented by its coordinates in the Euclidean space. Given a manipulation-related language command C (e.g., "Bring me the cup."), the final output affordance mask is $\boldsymbol{M} = [m_1, m_2, \ldots, m_n]$, where $m_i \in [0, 1]$. As Fig. 4 shows, our method uses a network that leverages a serial architecture: point cloud grouping, followed by semantic-aware feature alignment. The final fused feature is upsampled to the original shape and fed into a mask decoder to get query-related affordance segmentation.

$$\boldsymbol{M} = \mathbf{IDAS}(P|C), \tag{1}$$

where **IDAS**() denotes the proposed network.

4.2 Feature extraction

4.2.1 Point cloud feature

We use PointNet++ (Qi et al., 2017) for extracting features from point clouds. Specifically, we perform three layers of point set abstraction, in which the point cloud is progressively downsampled while the dimensionality of the features is increased. The original point cloud is represented as P. After downsampling, we obtain N key points defined as $P_{\mathrm{K}} = \{P_i^{\mathrm{K}}\}_{i=1}^{N}$. The encoded point feature after each downsampling stage is denoted as $\boldsymbol{F}_{\mathrm{point}} \in \mathbb{R}^{N \times D}$, where D represents the hidden dimensionality.

$$P_{\mathrm{K}} = \mathrm{DownSample}(P), \tag{2}$$

where DownSample() denotes the point set abstraction process.

4.2.2 Language feature

We begin by using a tokenizer to process the sentence, inputting it into a pre-trained language model. This approach helps capture the overall language context. Specifically, we use the uncased base version of robustly optimized pretraining for bidirectional encoder representation from Transformers (BERT) models (RoBERTa) (Liu YH et al., 2019) to obtain the sentence features, denoted as $\boldsymbol{F}_{\mathrm{text}} \in \mathbb{R}^{L \times d}$, where L represents the length of the input context sentence. The feature dimension d is set to 768, as defined by the RoBERTa-base-uncased model. For further calculations, we project the text features into a 512-dimensional space using a linear layer.

$$\boldsymbol{F}_{\mathrm{text}} = \boldsymbol{f}_{\mathrm{text}}(C), \tag{3}$$

where $\boldsymbol{f}_{\mathrm{text}}$ is the text encoder.

Since verbs and nouns play a crucial role in determining the meaning of user instructions, we introduce a VNA mechanism. We use spaCy (Montani et al., 2023) for part-of-speech tagging to extract a positional mask $\boldsymbol{M}_{\mathrm{vn}} \in \mathbb{R}^{L}$ for verbs and nouns. We design a verb–noun (VN)-encoder to encode the masked sentence as shown in Eq. (4):

$$\boldsymbol{F}_{\mathrm{vn}} = \tanh(\mathrm{LSTM}(\mathrm{LSTM}(\mathbf{embed}(C) * \boldsymbol{M}_{\mathrm{vn}}))), \tag{4}$$

where **embed**() denotes the embedding layer that converts C into a dense vector representation. $*$ denotes the element-wise multiplication between the embedded feature **embed**(C) and the mask $\boldsymbol{M}_{\mathrm{vn}}$.

The VN-encoder is a two-layer long short-term memory (LSTM) with tanh activation functions. It processes the input sentences token by token, maintaining a hidden state that evolves to capture contextual information. This approach allows the model to recognize word order, syntactic structures, and semantic relationships. The final hidden state, or a combination of states, provides a concise and informative representation of the sentence.

4.3 Multimodal fusion

Existing methods typically use point cloud feature extraction backbones that operate independently of the input text. This results in the generation of general-purpose features that are not specifically aligned with the query. To overcome this limitation, we introduce a QAM module that incorporates semantic information directly into the feature extraction process. Inspired by the work of Perez et al. (2017), we use language features to modulate the point cloud features. The design of the fusion module is depicted in Fig. 4b. The QAM module applies affine transformations to intermediate point cloud features based on previous language embeddings. This design allows the network to adaptively modulate its output according to the input command. The parameters α and β are learned from a multilayer perceptron (MLP). The final fused feature is represented as $\boldsymbol{F}_{\mathrm{fusion}} \in \mathbb{R}^{N \times D}$.

$$\alpha, \beta \leftarrow \mathrm{Linear}(\mathrm{ReLU}(\mathrm{Linear}(\boldsymbol{F}_{\mathrm{text}}))), \tag{5}$$

$$\mathrm{QAM}\left(\boldsymbol{F}_{\mathrm{point}} \mid \alpha, \beta\right) = \alpha \boldsymbol{F}_{\mathrm{point}} + \beta, \tag{6}$$

$$\boldsymbol{F}_{\mathrm{fusion}} = \mathrm{QAM}(\boldsymbol{F}_{\mathrm{text}}, \boldsymbol{F}_{\mathrm{point}}), \tag{7}$$

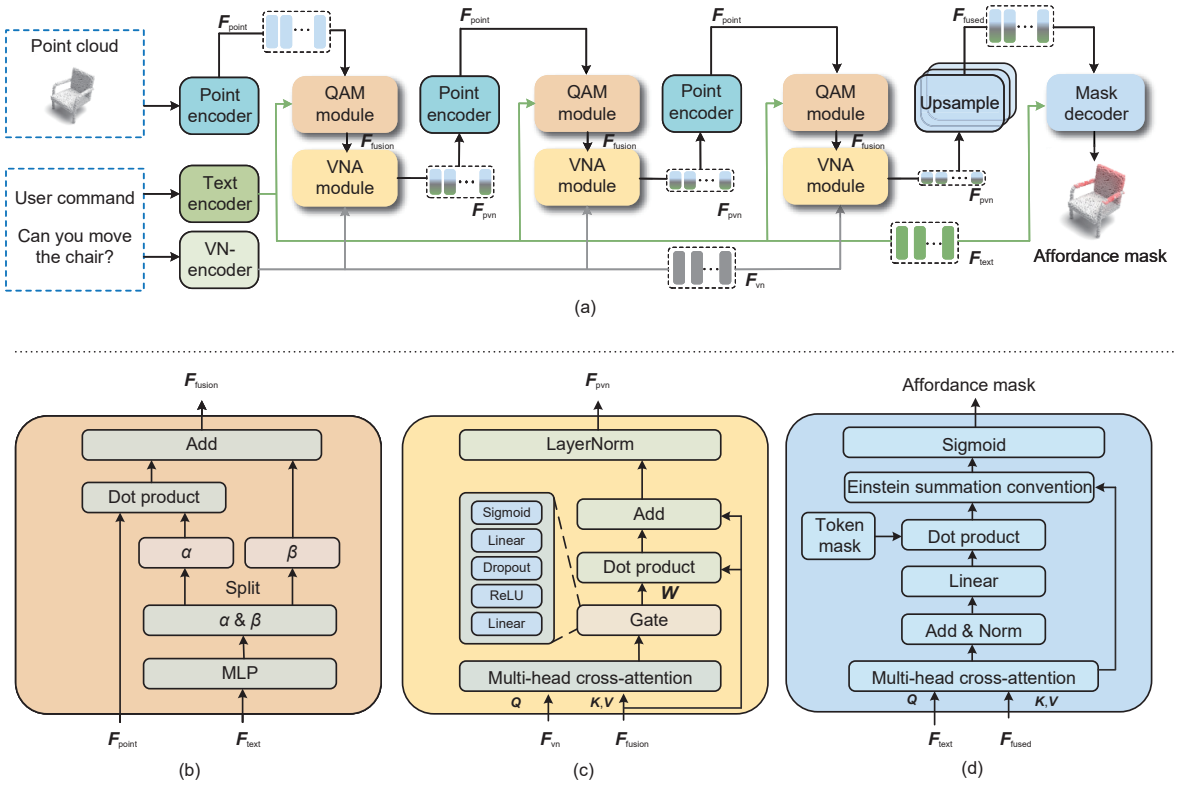

Fig. 4 IDAS network: (a) the overall architecture; (b) QAM module architecture; (c) VNA module architecture; (d) mask decoder architecture

where Linear() is the linear layer and rectified linear unit (ReLU) is the activation function.

To enable precise modulation of verbs and nouns in point cloud features, we have designed a VNA module (Fig. 4c).

This module uses multi-head cross-attention and learnable gating mechanism to selectively enhance or suppress individual point representations based on textual context. We begin by applying multi-head attention, using textual tokens as queries $\boldsymbol{Q}$ and point cloud features as keys $\boldsymbol{K}$ and values $\boldsymbol{V}$. This approach allows each textual token to focus on the entire set of 3D points.

$$\text{Attention}(\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V}) = \text{softmax}\left(\frac{\boldsymbol{Q}\boldsymbol{K}^{\mathrm{T}}}{\sqrt{d_{\mathrm{k}}}}\right)\boldsymbol{V}, \tag{8}$$

where Attention() is the attention mechanism, softmax() is the normalized exponential function, and d_{k} denotes the dimension of the key vector.

To obtain a compact global representation of the text-conditioned point features, we perform mean pooling over the sequence dimension. We concatenate each point feature with the corresponding context vector and pass it through a lightweight MLP to compute a gating weight:

$$\boldsymbol{W} = \sigma(\text{MLP}(\text{MHCA}(\boldsymbol{F}_{\mathrm{vn}}, \boldsymbol{F}_{\mathrm{fusion}}))), \tag{9}$$

where σ denotes the sigmoid activation function, and MHCA() denotes the multi-head cross-attention module for modeling the interaction between $\boldsymbol{F}_{\mathrm{vn}}$ and $\boldsymbol{F}_{\mathrm{fusion}}$. This gate learns to modulate the importance of each point conditioned on the textual context. The final output is computed by applying the gate and adding a residual connection, followed by layer normalization.

$$\boldsymbol{F}_{\mathrm{pvn}} = \text{LayerNorm}(\boldsymbol{W} * \boldsymbol{F}_{\mathrm{fusion}} + \boldsymbol{F}_{\mathrm{fusion}}), \tag{10}$$

where $\boldsymbol{F}_{\mathrm{pvn}}$ denotes the hybrid feature generated by the VNA module, and LayerNorm() denotes the layer normalization operation applied to normalize the output feature. This design allows the model to adaptively reweight point cloud features based on verb and noun semantics in an interpretable manner. The gating weights can also serve as a form of point-level attention map for visualizing the influence of text on 3D perception.

4.4 Mask decoder

We input the modulated fused feature into a Transformer decoder (Vaswani et al., 2017), along with the command feature. The decoder uses the question feature as the query and fused feature $\boldsymbol{F}_{\mathrm{fusion}}$ as both the key and the value, allowing it to output the alignment between the text and the point cloud. Next, we mask the invalid tokens in the query sentence and compute the dot-product similarity between each text token and each point. This produces a text–point semantic relevance map. We then sum the responses across all text tokens to generate an overall text-driven response for each point.

To normalize the results and avoid bias from varying text lengths, we divide the summed scores by the number of valid text tokens in each sample. The normalized scores are then constrained to the range of [0, 1], serving as the probability or confidence scores, as shown in Fig. 4d. The mask decoder is

described as follows:

$$\boldsymbol{M} = \sigma\Big(\text{MaskHead}\left(\text{MHCA}(\boldsymbol{Q} = \boldsymbol{F}_{\text{text}}, \boldsymbol{K}, \boldsymbol{V} = \boldsymbol{F}_{\text{fusion}})\right)\Big), \tag{11}$$

where MaskHead() denotes the mask prediction head used to generate the final affordance mask from the output of the MHCA module.

4.5 Loss function

Our model aims to learn a direct mapping between a user's command and the affordance, without being aware of any affordance categories. We formulate this as a binary classification task, where the goal is to predict the probability that each point corresponds to a specific potential affordance, with values ranging from 0 to 1. To address the class imbalance issue present in the IAD, we apply a combination of focal loss (Lin et al., 2017) and dice loss (Milletari et al., 2016). Here, $\mathcal{L}_{\text{Focal}}$ is the loss for point classification and $\mathcal{L}_{\text{Dice}}$ is the dice/F1 loss.

$$\mathcal{L}_{\text{Dice}} = 1 - \frac{2\sum_{i_{\text{s}}=1}^{M} p_{i_{\text{s}}} y_{i_{\text{s}}} + \epsilon}{\sum_{i_{\text{s}}=1}^{M} p_{i_{\text{s}}}^2 + \sum_{i_{\text{s}}=1}^{M} y_{i_{\text{s}}}^2 + \epsilon}, \tag{12}$$

$$\mathcal{L}_{\text{Focal}} = -\frac{1}{M}\sum_{i_{\text{s}}=1}^{M} \alpha_t (1-p_t)^{\gamma} \ln p_t, \tag{13}$$

$$p_t = \begin{cases} p_{i_{\text{s}}}, & \text{if } y_{i_{\text{s}}} = 1, \\ 1 - p_{i_{\text{s}}}, & \text{if } y_{i_{\text{s}}} = 0, \end{cases}$$

$$\alpha_t = \begin{cases} \alpha, & \text{if } y_{i_{\text{s}}} = 1, \\ 1 - \alpha, & \text{if } y_{i_{\text{s}}} = 0. \end{cases}$$

Here, $p_{i_{\text{s}}} \in [0,1]$ is the prediction probability of point i_{s}, while $y_{i_{\text{s}}} \in \{0,1\}$ is the ground truth of point i_{s}. M is the total number of supervised points in a batch. The term ϵ is a smoothing factor used to prevent division by zero, typically set to 1×10^{-5}. The focusing parameter γ controls how strongly well-classified examples are down-weighted, enabling the model to focus on hard samples. The final loss function is presented as

$$\mathcal{L}_{\text{Total}} = \omega\mathcal{L}_{\text{Dice}} + (1-\omega)\mathcal{L}_{\text{Focal}}. \tag{14}$$

Here, ω is a hyperparameter that controls the tradeoff between the focal loss and the dice loss. In Section 5.4.3, we discuss the suitable value of hyperparameter ω.

5 Experiments and results

5.1 Metrics

Following Li YC et al. (2024), we use four evaluation metrics include area under the curve (AUC), mean intersection over union (mIoU), similarity (SIM), and mean absolute error (MAE).

AUC is used to evaluate the model's ability to distinguish between affordance and non-affordance regions. This metric assesses the classification performance across multiple threshold levels, thereby quantifying the model's effectiveness in identifying relevant object parts under different scenarios. Higher AUC values correspond to better model performance. Next, mIoU is a standard evaluation metric for semantic segmentation tasks in computer vision. It measures the average overlap between the predicted segmentation masks and ground-truth labels across all classes. mIoU is the mean of the intersection over union (IoU) over all categories, with IoU expressed as follows:

$$\text{IoU} = \frac{\text{TP}}{\text{TP} + \text{FP} + \text{FN}}, \tag{15}$$

where TP, FP, and FN denote the numbers of true positives, false positives, and false negatives, respectively. SIM is a metric used to evaluate the similarity between the predicted mask and ground-truth masks. It is computed as

$$\begin{aligned} \text{SIM}\left(\boldsymbol{P}_{\text{m}}, \boldsymbol{Q}^{\text{D}}\right) &= \sum_{i=1}^{n} \min\left(P_{\text{m}i}, Q_i^{\text{D}}\right), \\ \sum_{i=1}^{n} P_{\text{m}i} &= \sum_{i=1}^{n} Q_i^{\text{D}} = 1, \end{aligned} \tag{16}$$

where $\boldsymbol{P}_{\text{m}}$ is the prediction map and $\boldsymbol{Q}^{\text{D}}$ denotes the continuous ground-truth distribution map. SIM computes the sum of the minimum values at each element, after normalizing the input maps. MAE is a widely used metric that measures the average magnitude of errors between the predicted and actual values.

$$\text{MAE} = \frac{1}{T}\sum_{i_{\text{s}}=1}^{T} \left|e_{i_{\text{s}}}\right|. \tag{17}$$

Here, $e_{i_{\text{s}}}$ is the calculated model error. MAE computes the total error by summing the absolute values of the errors and then dividing the result by the number of evaluated points T. We expect a low MAE score from our model.

5.2 Training details

We use the PyTorch framework (version 2.4 in a CUDA 12.1 environment) and Python 3.8 to implement our model. Our model is trained on a single NVIDIA RTX 4090 GPU with 24 GB of memory. The number of points for each object is initially fixed at 2048, and then downsampled to 512, 128, and 64 in a step-by-step manner. The hidden dimension of the final point and text features is set at 512. The model is trained for 50 epochs using the Adam optimizer with a batch size of 32 and a learning rate set to 1×10^{-4}.

5.3 Comparison with other methods

Our language-conditioned affordance segmentation is essentially a 3D visual grounding problem, akin to the referring expression segmentation (RES) task. To demonstrate the effectiveness of our method, we compare its performance against several baselines under identical training settings. We adopt two great 2D visual grounding networks, referring Transformer (ReferTrans) (Li MC and Sigal, 2021) and relationship modeling (ReLA) (Liu C et al., 2023), for comparison, substituting their image encoder with PointNet++, while leaving their fusion modules and mask decoder unchanged. Open-vocabulary affordance detection (OpenAD) (Nguyen et al., 2023) uses a similar input setting, so we retain their cosine similarity approach for aligning point features and text features. For the 3D single-stage referred point progressive selection (3D-SPS) (Luo et al., 2022) model, we remove its bounding box prediction module while retaining the other components. Interaction-driven 3D affordance grounding network (IAGNet) (Yang YH et al., 2023) is designed to learn 3D affordances with

interaction hints in 2D images. We replace its image backbone with a language model, while preserving the remainder of its architecture. Language-guided affordance segmentation on 3D object (LASO) (Li YC et al., 2024) can be trained directly using our dataset without any change.

5.4 Results

5.4.1 Seen and unseen

Table 2 demonstrates that our method achieves the best overall performance on the proposed task. In particular, our model attains the highest mIoU score of 21.25, outperforming the previous best-performing baseline LASO, which achieves an mIoU score of 18.88. Although the performance under the unseen setting drops compared to that under the seen setting, our method still achieves competitive results, demonstrating its ability to generalize beyond the training distribution. This suggests that the proposed model captures intrinsic affordance-related priors rather than merely memorizing object-specific patterns, thereby exhibiting effective knowledge transfer under unseen scenarios. Qualitative results are shown in Fig. 5. We also do zero-shot affordance segmentation using paraphrased instruction sentences that are not included in the training templates, as shown in Fig. 6. The results show that the model can correctly localize operable regions under such unseen instructions, demonstrating strong generalization to out-of-distribution language inputs.

Table 3 summarizes the performance of different affordance categories. We observe that affordances such as lift, stab, pull, and open consistently achieve strong results in both seen and unseen settings, particularly with respect to mIoU and AUC. A plausible explanation is that these affordances are characterized by relatively simple interaction patterns and well-defined operation regions, such as bag handles or door handles. As a result, the model can more easily align with instruction semantics with stable physical structures of objects. In contrast, affordances such as wear, push, and wrap-grasp exhibit noticeably weaker performance. These interactions typically lack clearly localized operation areas, and the valid interaction regions are often spatially broad or ambiguous. Such characteristics make it more challenging for the model to learn consistent physical priors that can be shared across different object instances. Furthermore, when comparing unseen scenes to seen scenes, affordances such as move, open, grasp, and pour show relatively smaller performance degradation. This suggests stronger generalization capability under distribution shifts. One likely reason is that these affordances are supported by a larger number of training samples and a greater diversity of object categories, enabling the model to learn more transferable patterns for grounding language instructions in spatial visual features.

5.4.2 Ablation study for the proposed modules

The ablation study presented in Table 4 systematically evaluates the contributions of the key components in our proposed IDAS model. Removing either the QAM module

Table 2 Performance comparison of different methods under seen and unseen settings in the IAD

Model	Seen				Unseen			
	mIoU↑	AUC↑	SIM↑	MAE↓	mIoU↑	AUC↑	SIM↑	MAE↓
3D-SPS	11.74	75.47	0.417	0.171	8.45	73.41	0.409	0.188
ReferTrans	12.74	78.47	0.487	0.164	9.88	74.85	0.451	0.178
ReLA	14.85	79.62	0.524	0.159	10.27	75.44	0.458	0.165
OpenAD	15.41	82.15	0.531	0.144	12.85	78.08	0.486	0.151
IAGNet	17.66	83.54	0.565	0.126	13.32	80.85	0.502	0.145
LASO	18.88	85.02	0.594	0.117	14.84	81.15	0.511	0.137
IDAS network (ours)	**21.25**	**85.62**	**0.614**	**0.101**	**15.74**	**82.37**	**0.529**	**0.127**

Bold values indicate the best performance among the compared methods. ↑ indicates the higher the better and ↓ indicates the lower the better

Table 3 Performance of the IDAS for each affordance type

Affordance type	Seen				Unseen			
	mIoU↑	AUC↑	SIM↑	MAE↓	mIoU↑	AUC↑	SIM↑	MAE↓
Move	14.41	78.90	0.587	0.136	10.05	70.39	0.543	0.178
Open	28.19	91.56	0.436	0.060	20.87	87.79	0.423	0.098
Grasp	22.75	82.76	0.603	0.114	18.03	77.97	0.548	0.159
Pour	22.29	89.77	0.659	0.088	18.78	80.65	0.590	0.135
Wrap-grasp	8.71	68.99	0.722	0.133	5.94	62.08	0.557	0.195
Press	19.30	93.70	0.496	0.048	15.47	91.11	0.481	0.092
Stab	37.88	98.27	0.549	0.024	27.88	93.80	0.497	0.076
Cut	16.34	93.54	0.759	0.075	9.61	91.26	0.671	0.102
Wear	7.92	68.65	0.626	0.148	6.87	69.27	0.597	0.173
Push	12.70	85.09	0.425	0.081	6.94	83.73	0.473	0.115
Pull	27.94	83.75	0.249	0.040	17.25	85.61	0.473	0.115
Lift	36.56	92.43	0.412	0.068	30.66	94.73	0.491	0.086

↑ indicates the higher the better and ↓ indicates the lower the better

Table 4 Ablation study of the proposed modules

Model	Seen				Unseen			
	mIoU↑	AUC↑	SIM↑	MAE↓	mIoU↑	AUC↑	SIM↑	MAE↓
w/o QAM module	19.55	85.17	0.606	0.106	15.18	82.35	0.517	0.132
w/o VNA module	19.87	85.23	0.608	0.110	15.29	82.06	0.516	0.130
w/o QAM and VNA modules	19.22	84.93	0.598	0.115	15.04	81.72	0.511	0.135
w/o VN-encoder	20.04	84.96	0.608	0.112	15.19	82.22	0.520	0.133
IDAS network	**21.25**	**85.62**	**0.614**	**0.101**	**15.74**	**82.37**	**0.529**	**0.127**

Bold values indicate the best performance among the compared settings. ↑ indicates the higher the better and ↓ indicates the lower the better

Fig. 5 Example results under seen and unseen settings

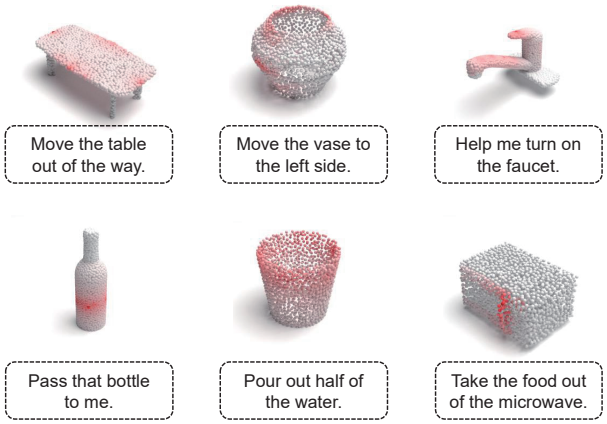

Fig. 6 Examples of instructions outside the IAD

or the VNA module leads to consistent performance degradation across all evaluation metrics. Specifically, removing the QAM module results in a noticeable decrease in mIoU and SIM, highlighting its critical role in aligning point cloud features with instruction semantics. Similarly, removing the VNA module causes a decline in precision, indicating that this module is effective in capturing fine-grained semantic cues from user commands. When both modules are removed, performance drops further, confirming the complementary nature of the QAM and VNA modules. In addition, we investigate the impact of the VN-encoder. As shown in Table 4, in the w/o VN-encoder setting (w/o denotes without), we replace the VN-encoder with the RoBERTa text encoder while retaining the VNA module. This modification results in a slight but consistent performance decrease, demonstrating that VN-encoder contributes meaningful visual–linguistic representations beyond generic text encoding. Overall, these results validate the effectiveness and necessity of each component in achieving a superior performance of the complete IDAS model.

5.4.3 Hyperparameter analysis

Our IDAS model was jointly trained using the sum of $\mathcal{L}_{\text{Dice}}$ and $\mathcal{L}_{\text{Focal}}$. After trying many different ratios of their combination, we find that the metric achieves the best result when $\mathcal{L}_{\text{Dice}} : \mathcal{L}_{\text{Focal}} = 1 : 1$, as shown in Table 5.

Table 5 Ablation study of the selection of the hyperparameter for the loss function

ω	Seen				Unseen			
	mIoU↑	AUC↑	SIM↑	MAE↓	mIoU↑	AUC↑	SIM↑	MAE↓
0.0	19.02	84.92	0.597	0.112	15.02	79.86	0.494	0.149
0.1	19.08	85.28	0.598	0.110	15.15	81.08	0.501	0.142
0.2	19.34	85.36	0.601	0.109	15.36	81.25	0.505	0.139
0.3	19.46	85.37	0.604	0.104	15.58	81.44	0.510	0.136
0.4	20.18	85.58	0.608	0.102	15.70	81.69	0.517	0.131
0.5	**21.25**	**85.62**	**0.614**	**0.101**	**15.74**	**82.37**	**0.529**	**0.127**
0.6	19.74	85.47	0.610	0.103	15.54	82.19	0.527	0.130
0.7	19.55	85.45	0.609	0.104	15.30	82.04	0.521	0.132
0.8	19.22	85.32	0.606	0.106	15.11	81.95	0.514	0.135
0.9	18.95	85.26	0.602	0.108	14.85	81.78	0.510	0.138
1.0	18.57	85.13	0.599	0.111	14.62	81.66	0.508	0.140

Bold values indicate the best performance among the compared settings. ↑ indicates the higher the better and ↓ indicates the lower the better

Table 6 Ablation study of different language backbones

Model	Seen				Unseen			
	mIoU↑	AUC↑	SIM↑	MAE↓	mIoU↑	AUC↑	SIM↑	MAE↓
DeBERTa	19.56	85.18	0.606	0.106	15.18	82.32	0.517	0.132
CLIP	19.87	85.26	0.608	0.110	15.25	82.09	0.516	0.130
RoBERTa	**21.25**	**85.62**	**0.614**	**0.101**	**15.74**	**82.37**	**0.529**	**0.127**

Bold values indicate the best performance among the compared methods. ↑ indicates the higher the better and ↓ indicates the lower the better

5.4.4 Different language encoders

As shown in Table 6, DeBERTa (He et al., 2020) and contrastive language–image pre-training (CLIP) (Radford et al., 2021) exhibit comparable performance, whereas RoBERTa (Liu YH et al., 2019) achieves much better results in both seen and unseen settings, which is consistent with the analysis result that RoBERTa is more suitable for our language-guided affordance segmentation task.

6 Conclusions

We present a novel task, instruction-guided affordance segmentation, which aims to detect the correct manipulation position on an object's 3D point cloud in an end-to-end manner, conditioned on user requirements. To support this task, we create a novel dataset, IAD, including instruction sentences and object point clouds with multiple affordance labels. Meanwhile, we propose the IDAS network, a deep learning framework that learns the map between user commands and latent affordance regions. Experiments on our IAD examples show that our method has great potential in both task-oriented robot manipulation and generalization to different instructions.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (No. 61973192). We thank the Shandong National Center for Applied Mathematics for data support.

Author contributions

Jiaxuan DU contributed to the concept and design of the study, data collection, data analysis, and drafting of the paper. Zhixian ZHAO contributed to data collection and analysis. Hao WU and Qing MA contributed to the critical revision of the paper. Guohui TIAN and Shuwen LENG supervised the study. Jiaxuan DU finalized the paper.

Conflict of interest

All the authors declare that they have no conflict of interest.

Data availability

The data that support the findings of this study are available from the corresponding author upon reasonable request.

Declaration on the use of generative AI tools

During the preparation of this work, the authors used ChatGPT to improve language. After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the content of the published article.

References

Achlioptas P, Abdelreheem A, Xia F, et al., 2020. Referit3D: neural listeners for fine-grained 3D object identification in real-world scenes. 16th European Conf on Computer Vision, p.422-440. https://doi.org/10.1007/978-3-030-58452-8_25

Ardón P, Pairet È, Petrick RPA, et al., 2019. Learning grasp affordance reasoning through semantic relations. *IEEE Robot Autom Lett*, 4(4):4571-4578. https://doi.org/10.1109/LRA.2019.2933815

Chen DZ, Chang AX, Nießner M, 2020. ScanRefer: 3D object localization in RGB-D scans using natural language. 16th European Conf on Computer Vision, p.202-221. https://doi.org/10.1007/978-3-030-58565-5_13

Deng SH, Xu X, Wu CZ, et al., 2021. 3D AffordanceNet: a benchmark for visual object affordance understanding. Proc IEEE/CVF Conf on Computer Vision and Pattern Recognition, p.1778-1787. https://doi.org/10.1109/CVPR46437.2021.00182

Devlin J, Chang MW, Lee K, et al., 2019. BERT: pre-training of deep bidirectional Transformers for language understanding. Proc Conf

of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, p.4171-4186. https://doi.org/10.18653/v1/N19-1423

Do TT, Nguyen A, Reid I, 2018. AffordanceNet: an end-to-end deep learning approach for object affordance detection. IEEE Int Conf on Robotics and Automation, p.5882-5889. https://doi.org/10.1109/icra.2018.8460902

Fang HS, Wang CX, Fang HJ, et al., 2023. AnyGrasp: robust and efficient grasp perception in spatial and temporal domains. *IEEE Trans Robot*, 39(5):3929-3945. https://doi.org/10.1109/TRO.2023.3281153

Fang K, Wu TL, Yang D, et al., 2018. Demo2Vec: reasoning object affordances from online videos. IEEE/CVF Conf on Computer Vision and Pattern Recognition, p.2139-2147. https://doi.org/10.1109/cvpr.2018.00228

Gibson JJ, 1978. The ecological approach to the visual perception of pictures. *Leonardo*, 11(3):227-235. https://doi.org/10.2307/1574154

Goyal M, Modi S, Goyal R, et al., 2022. Human hands as probes for interactive object understanding. Proc IEEE/CVF Conf on Computer Vision and Pattern Recognition, p.3283-3293. https://doi.org/10.1109/cvpr52688.2022.00329

He PC, Liu XD, Gao JF, et al., 2020. DeBERTa: decoding-enhanced BERT with disentangled attention. https://doi.org/10.48550/arXiv.2006.03654

Huang PH, Lee HH, Chen HT, et al., 2021. Text-guided graph neural networks for referring 3D instance segmentation. Proc 35th AAAI Conf on Artificial Intelligence, p.1610-1618. https://doi.org/10.1609/aaai.v35i2.16253

Islam R, Moushi OM, 2025. GPT-4o: the cutting-edge advancement in multimodal LLM. In: Arai K (Ed.), Intelligent Computing. Lecture Notes in Networks and Systems, Springer, Cham, p.47-60. https://doi.org/10.1007/978-3-031-92611-2_4

Li MC, Sigal L, 2021. Referring Transformer: a one-step approach to multi-task visual grounding. Proc 35th Int Conf on Neural Information Processing Systems, Article 1503.

Li YC, Zhao N, Xiao JB, et al., 2024. LASO: language-guided affordance segmentation on 3D object. Proc IEEE/CVF Conf on Computer Vision and Pattern Recognition, p.14251-14260. https://doi.org/10.1109/cvpr52733.2024.01351

Lin TY, Goyal P, Girshick R, et al., 2017. Focal loss for dense object detection. Proc IEEE Int Conf on Computer Vision, p.2999-3007. https://doi.org/10.1109/iccv.2017.324

Liu C, Ding HH, Jiang XD, 2023. GRES: generalized referring expression segmentation. IEEE/CVF Conf on Computer Vision and Pattern Recognition, p.23592-23601. https://doi.org/10.1109/cvpr52729.2023.02259

Liu SP, Tian GH, Cui YC, et al., 2022. A deep Q-learning network based active object detection model with a novel training algorithm for service robots. *Front Inform Technol Electron Eng*, 23(11):1673-1683. https://doi.org/10.1631/FITEE.2200109

Liu YH, Ott M, Goyal N, et al., 2019. RoBERTa: a robustly optimized BERT pretraining approach. https://arxiv.org/abs/1907.11692

Luo JY, Fu JH, Kong XH, et al., 2022. 3D-SPS: single-stage 3D visual grounding via referred point progressive selection. Proc IEEE/CVF Conf on Computer Vision and Pattern Recognition, p.16433-16442. https://doi.org/10.1109/cvpr52688.2022.01596

Milletari F, Navab N, Ahmadi SA, 2016. V-Net: fully convolutional neural networks for volumetric medical image segmentation. 4th Int Conf on 3D Vision, p.565-571. https://doi.org/10.1109/3dv.2016.79

Mo KC, Qin YZ, Xiang FB, et al., 2022. O2O-Afford: annotation-free large-scale object-object affordance learning. 5th Conf on Robot Learning, p.1666-1677.

Montani I, Honnibal M, Boyd A, et al., 2023. Explosion/spaCy: v3.7.2: Fixes for APIs and Requirements. Zenodo. https://doi.org/10.5281/zenodo.1212303 [Accessed on Feb. 1, 2026].

Mousavian A, Eppner C, Fox D, 2019. 6-DOF GraspNet: variational grasp generation for object manipulation. Proc IEEE/CVF Int Conf on Computer Vision, p.2901-2910. https://doi.org/10.1109/iccv.2019.00299

Nagarajan T, Feichtenhofer C, Grauman K, 2019. Grounded human-object interaction hotspots from video. Proc IEEE/CVF Int Conf on Computer Vision, p.8687-8696. https://doi.org/10.1109/iccv.2019.00878

Nguyen T, Vu MN, Vuong A, et al., 2023. Open-vocabulary affordance detection in 3D point clouds. IEEE/RSJ Int Conf on Intelligent Robots and Systems, p.5692-5698. https://doi.org/10.1109/iros55552.2023.10341553

Perez E, Strub F, De Vries H, et al., 2017. FiLM: visual reasoning with a general conditioning layer. 32nd AAAI Conf on Artificial Intelligence, p.3942-3951. https://doi.org/10.1609/aaai.v32i1.11671

Qi CR, Yi L, Su H, et al., 2017. PointNet++: deep hierarchical feature learning on point sets in a metric space. Proc 31st Int Conf on Neural Information Processing Systems, p.5105-5114.

Qian SY, Chen WF, Bai M, et al., 2024. AffordanceLLM: grounding affordance from vision language models. IEEE/CVF Conf on Computer Vision and Pattern Recognition Workshops, p.7587-7597. https://doi.org/10.1109/cvprw63382.2024.00754

Qin XF, Hu WK, Xiao C, et al., 2023. Attention-based efficient robot grasp detection network. *Front Inform Technol Electron Eng*, 24(10):1430-1444. https://doi.org/10.1631/FITEE.2200502

Radford A, Kim JW, Hallacy C, et al., 2021. Learning transferable visual models from natural language supervision. 38th Int Conf on Machine Learning, p.8748-8763.

Roh J, Desingh K, Farhadi A, et al., 2022. LanguageRefer: spatial-language model for 3D visual grounding. 5th Conf on Robot Learning, p.1046-1056.

Roy A, Todorovic S, 2016. A multi-scale CNN for affordance segmentation in RGB images. 14th European Conf on Computer Vision, p.186-201. https://doi.org/10.1007/978-3-319-46493-0_12

Song HO, Fritz M, Goehring D, et al., 2015. Learning to detect visual grasp affordance. *IEEE Trans Autom Sci Eng*, 13(2):798-809. https://doi.org/10.1109/tase.2015.2396014

Sundermeyer M, Mousavian A, Triebel R, et al., 2021. Contact-GraspNet: efficient 6-DoF grasp generation in cluttered scenes. IEEE Int Conf on Robotics and Automation, p.13438-13444. https://doi.org/10.1109/icra48506.2021.9561877

Vaswani A, Shazeer N, Parmar N, et al., 2017. Attention is all you need. Proc 31st Int Conf on Neural Information Processing Systems, p.6000-6010.

Wang Q, Fan Z, Sheng WH, et al., 2022. Cloud-assisted cognition adaptation for service robots in changing home environments. *Front Inform Technol Electron Eng*, 23(2):246-257. https://doi.org/10.1631/FITEE.2000431

Yang YH, Zhai W, Luo HC, et al., 2023. Grounding 3D object affordance from 2D interactions in images. IEEE/CVF Int Conf on Computer Vision, p.10871-10881. https://doi.org/10.1109/iccv51070.2023.01001

Yang ZY, Zhang SY, Wang LW, et al., 2021. SAT: 2D semantics assisted training for 3D visual grounding. Proc IEEE/CVF Int Conf on Computer Vision, p.1836-1846. https://doi.org/10.1109/iccv48922.2021.00187

Zhao LC, Cai DG, Sheng L, et al., 2021. 3DVG-Transformer: relation modeling for visual grounding on point clouds. Proc IEEE/CVF Int Conf on Computer Vision, p.2908-2917. https://doi.org/10.1109/iccv48922.2021.00292